

Robust estimation for mixtures of Gaussian factor analyzers, based on trimming and constraints

L.A. García-Escudero^a, A. Gordaliza^a, F. Greselin^b, S. Ingrassia^c, A. Mayo-Iscar^a

^a*Department of Statistics and Operations Research and IMUVA, University of Valladolid, Valladolid, Spain*

^b*Department of Statistics and Quantitative Methods, Milano-Bicocca University, Milan, Italy*

^c*Department of Economics and Business, University of Catania, Catania, Italy*

Abstract. Mixtures of Gaussian factors are powerful tools for modeling an unobserved heterogeneous population, offering - at the same time - dimension reduction and model-based clustering. Unfortunately, the high prevalence of spurious solutions and the disturbing effects of outlying observations, along maximum likelihood estimation, open serious issues. In this paper we consider restrictions for the component covariances, to avoid spurious solutions, and trimming, to provide robustness against violations of normality assumptions of the underlying latent factors. A detailed AECM algorithm for this new approach is presented. Simulation results and an application to the AIS dataset show the aim and effectiveness of the proposed methodology.

Keywords: Constrained estimation; Factor Analyzers Modeling; Mixture Models; Model-Based Clustering; Trimming; Robust estimation.

Contents

1	Introduction and motivation	2
2	Gaussian Mixtures of Factor Analyzers	4
3	Trimmed Mixtures of Factor Analyzers	5
3.1	Problem statement	5
3.2	Algorithm	6
4	Numerical studies	12
4.1	Artificial data	12
4.1.1	Properties of the estimators for the mixture parameters	15
4.2	Real data: the AIS data set	19

1 Introduction and motivation

Factor analysis is an effective method of summarizing the variability between a number of correlated features, through a much smaller number of unobservable, hence named *latent*, factors. It originated from the consideration that, in many phenomena, several observed variables could be explained by a few unobserved ones. Under this approach, each single observed variable (among the p ones) is assumed to be a linear combination of d underlying common factors with an accompanying error term to account for that part of variability that is unique to it (not in common with other variables). Ideally, d should be substantially smaller than p , to achieve parsimony.

Clearly, the effectiveness of this method is limited by its global linearity, as it happens for principal components analysis. Hence, Ghahramani and Hilton (1997), Tipping and Bishop (1999) and McLachlan and Peel (2000a) solidly widened the applicability of these approaches by combining local models of Gaussian factors in the form of finite mixtures. The idea is to employ latent variables to perform dimensional reduction in each component, thus providing a statistical method which concurrently performs clustering and, within each cluster, local dimensionality reduction.

In the literature, error and factors are routinely assumed to have a Gaussian distribution because of their mathematical and computational tractability: however statistical methods which ignore departure from normality may cause biased or misleading inference. Moreover, it is well known that maximum likelihood estimation for mixtures often leads to ill-posed problems because of the unboundedness of the objective function to be maximized, which favors the appearance of non-interesting local maximizers and degenerate or *spurious* solutions.

The lack of robustness in mixture fitting arises whenever the sample contains a certain proportion of data that does not follow the underlying population model. Spurious solutions can even appear when ML estimation is applied to artificial data drawn from a given finite mixture model, i.e. without adding any kind of contamination. Hence, robustified estimation is needed. Many contributions in this sense can be found in the literature: from Mclust model with a noise component in Fraley and Raftery (1998), mixtures of t -distributions in McLachlan and Peel (1998), the trimmed likelihood mixture fitting method in Neykov et al. (2007), the trimmed ML estimation of contaminated mixtures in Gallegos and Ritter (2009), and the robust improper

ML estimator introduced in Coretto and Hennig (2011), among many others. Some important applications in fields like computer vision, pattern recognition, analysis of microarray gene expression data, or tomography (see, for example, Stewart (1999), Campbell et al. (1997), Bickel (2003) and Maitra (2001), respectively) suggest that more attention should be paid to robustness, because noise in the data sets may be frequent in all these fields of application.

Different types of constraints have been traditionally applied in Gaussian mixtures of factor analyzers, for instance, some authors propose to take a common (diagonal) error matrix (as for the Mixtures of Common Factor Analyzers, denoted by MCFA, in Baek et al., 2010) or to impose an isotropic error matrix Bishop and Tipping, 1998. This strategy has proven to be effective in many cases, at the expenses of stronger distributional restrictions on the data. In McNicholas and Murphy, 2008, when analyzing parsimonious mixtures of Gaussians factor analyzers models, they realized that equal determinant restrictions give more stable results. For avoiding singularities and spurious solutions, under milder conditions, Greselin and Ingrassia (2013) recently proposed to maximize the likelihood by constraining the eigenvalues of the covariance matrices, following previous work of Ingrassia (2004) and going back to Hathaway (1985). Furthermore, McLachlan and Bean (2005), Baek and McLachlan (2011), Steane et al. (2012) and Lin et al. (2014) have considered the use of mixtures of t -analyzers in an attempt to make the model less sensitive to outliers, but they, too, are not robust against very extreme outliers (Hennig, 2004); while Fokoué and Titterton (2003) proposed a Bayesian approach.

The purpose of the present work is to introduce an estimating procedure for mixture of Gaussian factors analyzers that can resist the effect of outliers and avoid spurious local maximizers. The proposed constraints can be also used to take into account prior information about the scatter parameters.

Trimming has been shown to be a simple, powerful, flexible and computationally feasible way to provide robustness in many different statistical frameworks. The basic idea behind trimming here is the removal of a little proportion α of observations whose values would be the more unlikely to occur if the fitted model was true. In this way, trimming avoids that a small fraction of outlying observations could exert a harmful effect on the estimation of the parameters of the fitted model.

Incorporating constraints in the mixture fitting estimation method moves the mathematical problem in a well-posed setting and hence minimizes the risk of incurring spurious solutions. Moreover, a correct statement of the problem allows to study the properties of the EM algorithms as in Ingrassia and Rocci (2007) and to obtain the desired statistical properties for the

estimators, such as the existence and consistency results, as in García-Escudero et al. (2008) and Gallegos and Ritter (2009).

We have organized the rest of the paper as follows. In Section 2 we introduce notation and summarize main ideas about Gaussian Mixtures of Factor Analyzers (in the foremost, denoted by MFA). Then, in Section 3 we introduce the trimmed likelihood for MFA and we provide fairly extensive notes concerning the EM algorithm, with incorporated trimming and constrained estimation. In Section 4 we discuss the performance of our procedure, on the ground of some numerical results obtained from simulated and real data. In particular, we compare the bias and MSE of robustly estimated model parameters for different cases of data contamination, by Monte Carlo experiments. The application to the Australian Institute of Sports dataset shows how classification and factor analysis can be developed using the new model. Section 5 contains concluding notes and provides ideas for further research.

2 Gaussian Mixtures of Factor Analyzers

The density of the p -dimensional random variable \mathbf{X} of interest is modeled as a mixture of G multivariate normal densities in some unknown proportions π_1, \dots, π_G , whenever each data point is taken to be a realization of the following density function,

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sum_{g=1}^G \pi_g \phi_p(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \quad (2.1)$$

where $\phi_p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the p -variate normal density function with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Here the vector $\boldsymbol{\theta} = \boldsymbol{\theta}_{GM}(p, G)$ of unknown parameters consists of the $(G - 1)$ mixing proportions π_g , the Gp elements of the component means $\boldsymbol{\mu}_g$, and the $\frac{1}{2}Gp(p+1)$ distinct elements of the component-covariance matrices $\boldsymbol{\Sigma}_g$. MFA postulates a finite mixture of linear sub-models for the distribution of the full observation vector \mathbf{X} , given the (unobservable) factors \mathbf{U} . That is, MFA provides local dimensionality reduction by assuming that the distribution of the observation \mathbf{X}_i can be given as

$$\mathbf{X}_i = \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \mathbf{U}_{ig} + \mathbf{e}_{ig} \quad \text{with probability} \quad \pi_g \quad (g = 1, \dots, G) \quad \text{for } i = 1, \dots, n, \quad (2.2)$$

where $\boldsymbol{\Lambda}_g$ is a $p \times d$ matrix of *factor loadings*, the *factors* $\mathbf{U}_{1g}, \dots, \mathbf{U}_{ng}$ are $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ distributed independently of the *errors* \mathbf{e}_{ig} . The latter are independently $\mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_g)$ distributed, and $\boldsymbol{\Psi}_g$ is a $p \times p$ diagonal matrix ($g = 1, \dots, G$). The diagonality of $\boldsymbol{\Psi}_g$ is one of the key assumptions of factor analysis: the observed variables are independent given the factors. Note that the factor

variables \mathbf{U}_{ig} model correlations between the elements of \mathbf{X}_i , while the errors \mathbf{e}_{ig} account for independent noise for \mathbf{X}_i . We suppose that $d < p$, which means that d unobservable factors are jointly explaining the p observable features of the statistical units. Under these assumptions, the mixture of factor analyzers model is given by (2.1), where the g -th component-covariance matrix Σ_g has the form

$$\Sigma_g = \Lambda_g \Lambda_g' + \Psi_g \quad (g = 1, \dots, G). \quad (2.3)$$

The parameter vector $\boldsymbol{\theta} = \boldsymbol{\theta}_{MFA}(p, d, G)$ now consists of the elements of the component means $\boldsymbol{\mu}_g$, the Λ_g , and the Ψ_g , along with the mixing proportions π_g ($g = 1, \dots, G - 1$), on putting $\pi_G = 1 - \sum_{i=1}^{G-1} \pi_g$.

3 Trimmed Mixtures of Factor Analyzers

In this section we present the *trimmed (Gaussian) mixtures of factor analyzers model* (trimmed MFA) and we propose a feasible algorithm for its implementation.

3.1 Problem statement

We will fit a mixture of Gaussian factor components to a given dataset $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ in \mathbb{R}^p by maximizing a *trimmed mixture log-likelihood* (see Neykov et al. 2007, Gallegos and Ritter 2009 and García-Escudero et al. 2014) defined as:

$$\mathcal{L}_{trim} = \sum_{i=1}^n z(\mathbf{x}_i) \log \left[\sum_{g=1}^G \phi_p(\mathbf{x}_i; \boldsymbol{\mu}_g, \Lambda_g \Lambda_g' + \Psi_g) \pi_g \right] \quad (3.1)$$

where $z(\cdot)$ is a 0-1 trimming indicator function that tell us whether observation \mathbf{x}_i is trimmed off: $z(\mathbf{x}_i)=0$, or not: $z(\mathbf{x}_i)=1$ and $\Sigma_g = \Lambda_g \Lambda_g' + \Psi_g$ as in (2.3). A fixed fraction α of observations can be unassigned by setting $\sum_{i=1}^n z(\mathbf{x}_i) = [n(1 - \alpha)]$. Hence the parameter α denotes the trimming level. As usual, $\mathbf{x}_1, \dots, \mathbf{x}_n$ are the realized values of n independent and identically distributed random vectors $\mathbf{X}_1, \dots, \mathbf{X}_n$ with common density given in (2.1), with component-covariance matrices Σ_g as in (2.3) for $g = 1, \dots, G$. The component label vectors $\mathbf{z}_1, \dots, \mathbf{z}_n$ are taken to be the realized values of the random vectors $\mathbf{Z}_1, \dots, \mathbf{Z}_n$, where, for independent feature data, it is appropriate to assume that they are (unconditionally) multinomially distributed. i.e. $\mathbf{Z}_1, \dots, \mathbf{Z}_n \sim^{i.i.d.} Mult_G(1; \pi_1, \dots, \pi_G)$.

Moreover, to avoid the unboundedness of \mathcal{L}_{trim} , we introduce a *constrained maximization* of (3.1). In more detail, with reference to the diagonal elements $\{\psi_{g,k}\}_{k=1,\dots,p}$ of the noise matrices Ψ_g for $g = 1, \dots, G$ we require that

$$\psi_{g_1,k} \leq c_{noise} \psi_{g_2,h} \quad \text{for every } 1 \leq k \neq h \leq p \text{ and } 1 \leq g_1 \neq g_2 \leq G \quad (3.2)$$

The constant c_{noise} is finite and such that $c_{noise} \geq 1$, to avoid the $|\Sigma_g| \rightarrow 0$ case. This constraint can be seen as an adaptation to MFA of those introduced in Ingrassia and Rocci (2007), García-Escudero et al. (2008) and it is similar to the mild restrictions implemented for MFA in Greselin and Ingrassia (2013). They all go back to the seminal paper of Hathaway (1985). We will look for the ML estimators of Ψ_g under the given constraints, and this position set the maximization problem as a well-defined one, and at the same time discard singularities and reduce spurious solutions.

If $\{\lambda_k(A)\}_{k=1,\dots,p}$ denote the set of eigenvalues of the $p \times p$ matrix A , a second set of constraints apply on the product of the loading matrices $\Lambda_g \Lambda'_g$ by requiring that

$$\lambda_k(\Lambda_{g_1} \Lambda'_{g_1}) \leq c_{load} \lambda_h(\Lambda_{g_2} \Lambda'_{g_2}) \text{ for every } 1 \leq k \neq h \leq d \text{ and } 1 \leq g_1 \neq g_2 \leq G. \quad (3.3)$$

with c_{load} such that $1 \leq c_{load} < +\infty$. These $\lambda_k(\Lambda_g \Lambda'_g)$ values control the different scatters in the reduced subspaces. In fact, these type of constraints are not needed to avoid singularities in the target function but they could be useful to achieve more sensible solutions.

In the foremost, we will denote by Θ_c the constrained parameter space for $\theta = \{\pi_g, \mu_g, \Psi_g, \Lambda_g; g = 1, \dots, G\}$ under the requirements (3.2) and (3.3).

3.2 Algorithm

The maximization of \mathcal{L}_{trim} in (3.1) for $\theta \in \Theta_c$ is not an easy task, obviously. We will give a feasible algorithm obtained by combining the Alternating Expectation-Conditional Maximization algorithm (AECM) for MFA with that (with trimming and constraints) introduced in García-Escudero et al. (2014) (see, also, Fritz et al., 2013).

The AECM is an extension of the EM, suggested by the factor structure of the model, that uses different specifications of missing data at each stage. The idea is to partition the vector of parameters $\theta = (\theta'_1, \theta'_2)'$ in such a way that \mathcal{L}_{trim} is easy to be maximized for θ_1 given θ_2 and viceversa, replacing the M-step by a number of computationally simpler conditional maximization (CM) steps. In more detail, in the first cycle we set $\theta_1 = \{\pi_g, \mu_g; g = 1, \dots, G\}$ and the missing data are the unobserved group labels $\mathbf{Z} = (\mathbf{z}'_1, \dots, \mathbf{z}'_n)$, while in the second cycle we set

$\theta_2 = \{\Lambda_g, \Psi_g; g = 1, \dots, G\}$ and the missing data are the group labels \mathbf{Z} and the unobserved latent factors $\mathbf{U} = (\mathbf{U}_{11}, \dots, \mathbf{U}_{nG})$. Hence, the application of the AECM algorithm consists of two cycles, and there is one E-step and one CM-step alternatively considering θ_1 and θ_2 in each cycle. Before describing the algorithm, we remark that the unobserved group labels \mathbf{Z} are considered missing data in both cycles. Therefore, during the l -th iteration, we shall denote by $z_{ig}^{(l+1/2)}$ and $z_{ig}^{(l+1)}$ the conditional expectations at the first and second cycle, respectively.

The algorithm has to be run multiple times on the same dataset, with different starting values, to prevent the attainment of a local, rather than global, maximum log-likelihood. In each run it executes the following steps:

1 Initialization:

Each iteration begins by selecting initial values for $\theta^{(0)}$ where $\theta^{(0)} = (\pi_g^{(0)}, \mu_g^{(0)}, \Lambda_g^{(0)}, \Psi_g^{(0)}; g = 1, \dots, G)$. Inspired from results obtained in a series of extensive test experiments about initialization strategies (see Maitra, 2009), and aiming to allow the algorithm to visit the entire parameter space, we randomly select $p + 1$ units (without replacement) for group g from the observed data $\mathbf{x} = \{\mathbf{x}_i\}_{i=1, \dots, n}$. In this way we obtain a subsample \mathbf{X}^g that we arrange in a $(p + 1) \times p$ matrix, and its sample mean will be the initial $\mu_g^{(0)}$. Additionally, based on these $p + 1$ observations, we developed a new *ad hoc* approach for providing an initialization procedure for $\Psi_g^{(0)}$ and $\Lambda_g^{(0)}$, to deal with the possible existence of gross outlying observations among the subsamples, which could inflate disproportionately some of their eigenvalues. The rationale under our procedure is, as usual, to fill in randomly the missing information in the complete model through random subsamples and, then, to estimate the other parameters. The missing information here are the factors $\mathbf{u}_1, \dots, \mathbf{u}_n$, which, under the assumptions for the model, are independently $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ distributed. We consider model (2.2) in group g as a regression of \mathbf{X}_i with intercept μ_g , regression coefficients given by Λ_g , where the explanatory variables are the latent factors \mathbf{U}_{ig} , and with regression errors \mathbf{e}_{ig} . Hence we draw $p + 1$ random observations from the d -variate standard Gaussian to fill a $(p + 1) \times d$ matrix \mathbf{U}^g . Then we set $\Lambda_g^{(0)} = (\mathbf{U}_g' \mathbf{U}_g)^{-1} \mathbf{U}_g' \mathbf{X}_c^g$ where \mathbf{X}_c^g is obtained by centering the columns of the \mathbf{X}^g matrix. To provide a restricted random generation of Ψ_g , we compute the $(p + 1) \times p$ matrix $\varepsilon_g = \mathbf{x}_c^g - \Lambda_g^{(0)} \mathbf{U}_g$, and we set the diagonal elements of $\Psi_g^{(0)}$ equal to the variances of the p columns of the ε_g matrix. We repeat this for $g = 1, \dots, G$ and if the obtained matrices

$\Lambda_g^{(0)}$ and $\Psi_g^{(0)}$ do not satisfy the required constraints (3.2) and (3.3), then the constrained maximizations described in step 2.4 must be applied. Finally, weights $\pi_1^{(0)}, \dots, \pi_G^{(0)}$ in the interval $(0, 1)$ and summing up to 1 are randomly chosen.

2 Trimmed AECM steps:

The following steps 2.1–2.4. are alternatively executed until convergence (i.e. $\|\mathbf{z}^{(l+1)} - \mathbf{z}^{(l)}\| < \epsilon$) for a small constant $\epsilon > 0$ or until reaching a maximum number of iterations *MaxIter*. The implementation of trimming is related to the “concentration” steps applied in high-breakdown robust methods (Rousseeuw and Van Driessen, 1999). Trimming is performed along the E-steps, while constraints are enforced during the second cycle CM step.

2.1 First cycle. E-step:

Here $\theta_1 = \{\pi_g, \mu_g; g = 1, \dots, G\}$ and the missing data are the unobserved group labels $\mathbf{z} = (\mathbf{z}'_1, \dots, \mathbf{z}'_n)$. The E-step on the first cycle on the $(l + 1)$ -th iteration requires the calculation of

$$Q_1(\theta_1; \theta^{(l)}) = \mathbb{E}_{\theta^{(l)}} [\mathcal{L}_{trim}(\theta_1) | \mathbf{x}],$$

which is the expected trimmed complete-data log-likelihood given the data \mathbf{x} and using the current estimate $\theta^{(l)}$ for θ . In practice it requires calculating $\mathbb{E}_{\theta^{(l)}} [Z_{ig} | \mathbf{x}]$ and usual computations show that this step is achieved by replacing each z_{ig} by its current conditional expectation given the observed data \mathbf{x}_i , that is we replace z_{ig} by $\tau_{ig}^{(l+1/2)}$, where the latter is evaluated as follows. Let us define

$$D_g(\mathbf{x}; \theta^{(l)}) = \phi_p(\mathbf{x}; \mu_g^{(l)}, \Lambda_g^{(l)} [\Lambda_g^{(l)}]^{-1} + \Psi_g^{(l)}) \pi_g^{(l)}$$

and

$$D_i = D(\mathbf{x}_i; \theta^{(l)}) = \sum_{g=1}^G D_g(\mathbf{x}_i; \theta^{(l)}), \text{ for } i = 1, \dots, n.$$

After sorting these n values, the notation $D_{(1)} \leq \dots \leq D_{(n)}$ is adopted. Let us consider the subset of indices $I \subset \{1, 2, \dots, n\}$ defined as

$$I = \{i : D_{(i)} \geq D_{([n\alpha])}\}. \quad (3.4)$$

To update the parameters, only the observations with indices in I will be taken into account. In other words, we are tentatively discarding the proportion α of observations with the smallest $D_{(i)}$ values.

Then, set

$$\tau_{ig}^{(l+1/2)} = \begin{cases} \frac{D_g(\mathbf{x}_i; \theta^{(l)})}{D(\mathbf{x}_i; \theta^{(l)})} & \text{for } i \in I \\ 0 & \text{for } i \notin I. \end{cases}$$

Note that, for the observations with indices in I , $\tau_{ig}^{(l+1/2)}$ are the ‘‘posterior probabilities’’ often considered in standard EM algorithms applied when fitting MFAs. But, unlike the standard EM algorithms, the $\tau_{ig}^{(l+1/2)}$ (and consequently the z_{ig}) for the discarded observations are set to 0.

2.2 First cycle. CM-step: This first CM step requires the maximization of $Q_1(\theta_1; \theta^{(l)})$ over θ , with θ_2 held fixed at $\theta_2^{(l)}$. We get $\theta_1^{(l+1)}$ by updating π_g and μ_g as follows

$$\pi_g^{(l+1)} = \frac{\sum_{i=1}^n \tau_{ig}^{(l+1/2)}}{[n(1 - \alpha)]}$$

and

$$\mu_g^{(l+1)} = \frac{\sum_{i=1}^n \tau_{ig}^{(l+1/2)} \mathbf{x}_i}{n_g^{(l+1/2)}}$$

where $n_g^{(l+1/2)} = \sum_{i=1}^n \tau_{ig}^{(l+1/2)}$, for $g = 1, \dots, G$. According to notation in McLachlan and Peel (2000b), we set $\theta^{(l+1/2)} = (\theta_1^{(l+1)}, \theta_2^{(l)})'$.

2.3 Second cycle. E- step:

Here we consider $\theta_2 = \{(\Lambda_g, \Psi_g), g = 1, \dots, G\}$, where the missing data are the unobserved group labels \mathbf{Z} and the latent factors \mathbf{U} . Therefore, the trimmed complete-data log-likelihood in this second cycle may be written as

$$\mathcal{L}_{trim:2}(\theta_2) = \sum_{i=1}^n z(\mathbf{x}_i) \log \sum_{g=1}^G [\phi_p(\mathbf{x}_i; \mu_g^{(l+1)} - \Lambda_g \mathbf{u}_{ig}, \Psi_g) \phi_d(\mathbf{u}_{ig}; 0, \mathbf{I}_d) \pi_g^{(l+1)}].$$

The E-step on the second cycle on the l -th iteration requires the calculation of the conditional expectation of $\mathcal{L}_{trim:2}$, given the observed data \mathbf{x} and using the current estimate $\theta^{(l+1/2)}$ for θ , i.e.

$$Q_2(\theta_2; \theta^{(l+1/2)}) = \mathbb{E}_{\theta^{(l+1/2)}} [\mathcal{L}_{trim:2}(\theta_2) | \mathbf{x}].$$

In addition to updating the posterior probabilities $\mathbb{E}_{\boldsymbol{\theta}^{(l+1/2)}}[Z_{ig}|\mathbf{x}]$ by performing a concentration step and replacing each z_{ig} by the new values $z_{ig}^{(l+1)} = \tau_{ig}^{(l+1)}$ (and consequently $n_g^{(l+1)} = \sum_{i=1}^n \tau_{ig}^{(l+1)}$, for $g = 1, \dots, G$, as previously done in step 2.1), this leads to evaluate the following conditional expectations: $\mathbb{E}_{\boldsymbol{\theta}^{(l+1/2)}}[Z_{ig}\mathbf{U}_{ig}|\mathbf{x}]$ and $\mathbb{E}_{\boldsymbol{\theta}^{(l+1/2)}}[Z_{ig}\mathbf{U}_{ig}\mathbf{U}_{ig}'|\mathbf{x}]$. Recalling that the conditional distribution of \mathbf{U}_{ig} given \mathbf{x}_i is

$$\mathbf{U}_{ig}|\mathbf{x}_i \sim \mathcal{N}(\gamma_g(\mathbf{x}_i - \boldsymbol{\mu}_g), \mathbf{I}_q - \gamma_g\boldsymbol{\Lambda}_g)$$

for $i = 1, \dots, n$ and $g = 1, \dots, G$ with

$$\gamma_g = \boldsymbol{\Lambda}_g'(\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g)^{-1},$$

we obtain

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\theta}^{(l+1/2)}}[Z_{ig}\mathbf{U}_{ig}|\mathbf{x}_i] &= z_{ig}^{(l+1)}\gamma_g^{(l)}(\mathbf{x}_i - \boldsymbol{\mu}_g^{(l+1)}) \\ \mathbb{E}_{\boldsymbol{\theta}^{(l+1/2)}}[Z_{ig}\mathbf{U}_{ig}\mathbf{U}_{ig}'|\mathbf{x}_i] &= z_{ig}^{(l+1)}\left[\gamma_g^{(l)}(\mathbf{x}_i - \boldsymbol{\mu}_g^{(l+1)})(\mathbf{x}_i - \boldsymbol{\mu}_g^{(l+1)})'\gamma_g^{(l)'} + \mathbf{I}_q - \gamma_g^{(l)}\boldsymbol{\Lambda}_g^{(l)}\right] \\ &= z_{ig}^{(l+1)}\boldsymbol{\Xi}_{ig}^{(l)},\end{aligned}$$

where we set

$$\begin{aligned}\gamma_g^{(l)} &= \boldsymbol{\Lambda}_g^{(l)'}\left(\boldsymbol{\Lambda}_g^{(l)}\boldsymbol{\Lambda}_g^{(l)'} + \boldsymbol{\Psi}_g^{(l)}\right)^{-1} \\ \boldsymbol{\Xi}_{ig}^{(l)} &= \mathbf{I}_q - \gamma_g^{(l)}\boldsymbol{\Lambda}_g^{(l)} + \gamma_g^{(l+1)}(\mathbf{x}_i - \boldsymbol{\mu}_g^{(l+1)})(\mathbf{x}_i - \boldsymbol{\mu}_g^{(l+1)})'\gamma_g^{(l)'}.\end{aligned}$$

2.4 Second cycle. CM-step for constrained estimation of $\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Psi}_g$:

Here our aim is to maximize $Q_2(\boldsymbol{\theta}_2; \boldsymbol{\theta}^{(l)})$ over $\boldsymbol{\theta}$, with $\boldsymbol{\theta}_1$ held fixed at $\boldsymbol{\theta}_1^{(l+1)}$. After some matrix algebra, this yields the updated ML-estimates

$$\begin{aligned}\boldsymbol{\Lambda}_g &= \mathbf{S}_g^{(l+1)}\gamma_g^{(l)'}[\boldsymbol{\Xi}_g^{(l)}]^{-1} \\ \boldsymbol{\Psi}_g &= \text{diag}\left\{\mathbf{S}_g^{(l+1)} - \boldsymbol{\Lambda}_g^{(l+1)}\gamma_g^{(l)}\mathbf{S}_g^{(l+1)}\right\}\end{aligned}$$

where we denote by $\mathbf{S}_g^{(l+1)}$ the sample scatter matrix in group g , for $g = 1, \dots, G$

$$\mathbf{S}_g^{(l+1)} = (1/n_g^{(l+1)})\sum_{i=1}^n z_{ig}^{(l+1)}(\mathbf{x}_i - \boldsymbol{\mu}_g^{(l+1)})(\mathbf{x}_i - \boldsymbol{\mu}_g^{(l+1)})'.$$

Along the iterations, due to the updates, it may happen that the $\boldsymbol{\Lambda}_g$ matrices do not belong to the constrained parameter space $\boldsymbol{\Theta}_c$. In case that the eigenvalues of

the $\Lambda_g \Lambda'_g$ matrices do not satisfy the required constraint (3.3), we obtain the ML solution in Θ_c by projecting the unconstrained optimum into Θ_c . To this aim, the singular-value decomposition of $\Lambda_g \Lambda'_g = \mathbf{L}'_g \mathbf{E}_g \mathbf{L}_g$ is considered, with \mathbf{L}_g being an orthogonal matrix and $\mathbf{E}_g = \text{diag}(e_{g1}, e_{g2}, \dots, e_{gd})$ a diagonal matrix (notice that some of these e_{gk} may be equal to 0 if $\Lambda_g \Lambda'_g$ is not full rank). Truncated singular values are then defined as

$$[e_{gk}]_m = \min(c_{load} \cdot m, (\max(e_{gk}, m))), \text{ for } k = 1, \dots, d \text{ and } g = 1, \dots, G,$$

and m being some threshold value. The loading matrices are finally updated as $\Lambda_g^{(l+1)}$ such that $\Lambda_g^{(l+1)} [\Lambda_g^{(l+1)}]' = \mathbf{L}'_g \mathbf{E}_g^* \mathbf{L}_g$ with

$$\mathbf{E}_g^* = \text{diag}([e_{g1}]_{m_{\text{opt}}}, [e_{g2}]_{m_{\text{opt}}}, \dots, [e_{gd}]_{m_{\text{opt}}})$$

and m_{opt} minimizing the real valued function

$$f_{load}(m) = \sum_{g=1}^G \pi_g^{(l+1)} \sum_{k=1}^d \left(\log([e_{gk}]_m) + \frac{e_{gk}}{[e_{gk}]_m} \right). \quad (3.5)$$

It may be mentioned here, in passing, that Proposition 3.2 in Fritz et al. (2013) shows that m_{opt} can be obtained by evaluating $2dG + 1$ times the real valued function $f_{load}(m)$ in (3.5).

Given the $\Lambda_g^{(l+1)}$, we obtain the matrices

$$\Psi_g = \text{diag} \left\{ \mathbf{S}_g^{(l+1)} - \Lambda_g^{(l+1)} \gamma_g^{(l)} \mathbf{S}_g^{(l+1)} \right\}$$

which may not necessarily satisfy the required constraint (3.2). In this case, we set

$$[\psi_{g,k}]_m = \min(c_{noise} \cdot m, \max(\psi_{g,l}, m)), \text{ for } k = 1, \dots, d; g = 1, \dots, G,$$

and fix the optimal threshold value m_{opt} by minimizing the following real valued function

$$f_{noise}(m) \mapsto \sum_{g=1}^G \pi_g^{(l+1)} \sum_{k=1}^p \left(\log([\psi_{g,k}]_m) + \frac{\psi_{g,k}}{[\psi_{g,k}]_m} \right). \quad (3.6)$$

As before, in Fritz et al. (2013) it is shown that m_{opt} can be obtained in a straightful way by evaluating $2pG + 1$ times $f_{noise}(m)$ in (3.6). Thus, $\Psi_g^{(l+1)}$ is finally updated as

$$\Psi_g^{(l+1)} = \text{diag}([\psi_{g,1}]_{m_{\text{opt}}}, \dots, [\psi_{g,p}]_{m_{\text{opt}}}).$$

It is worth to remark that the given constrained estimation provides, at each step, the parameters Ψ_g and Λ_g maximizing the likelihood in the constrained parameter space Θ_c .

3 *Evaluate target function:* After applying the trimmed and constrained EM steps, and setting $z(\mathbf{x}_i) = 0$ if $i \in I$ and $z(\mathbf{x}_i) = 1$ if $i \notin I$, the associated value of the target function (3.1) is evaluated. If convergence has not been achieved before reaching the maximum number of iterations $MaxIter$, results are discarded.

The set of parameters yielding the highest value of the target function (among the multiple runs) and the associated trimmed indicator function z are returned as the final output of the algorithm. In the framework of model-based clustering, each unit is assigned to one group, based on the maximum a posteriori probability. Notice, in passing, that we do not need a high number of initializations neither a high value for $MaxIter$, as we will see in Section 4.

4 Numerical studies

In this section we present numerical studies, based on simulated and real data, to show the performance of the constrained and trimmed AECM algorithm with respect to unconstrained and/or untrimmed approaches.

4.1 Artificial data

We consider here the following mixture of G components of d -variate normal distributions. To perform the estimation, we consider 10 different random initial clusterings to initialize the algorithm at each run, as described in the previous section, and we retain the best solution. The needed routines have been written in R-code (R Team, 2013), and are available from the authors upon request.

MIXTURE: $G = 3, d = 6, q = 2, N = 150$.

The sample has been generated with weights $\pi = (0.3, 0.4, 0.3)'$ according to the following parameters:

$$\begin{aligned} \mu_1 &= (0, 0, 0, 0, 0, 0)' & \Psi_1 &= \text{diag}(0.1, 0.1, 0.1, 0.1, 0.1, 0.1) \\ \mu_2 &= (5, 5, 5, 5, 5, 5)' & \Psi_2 &= \text{diag}(0.4, 0.4, 0.4, 0.4, 0.4, 0.4) \\ \mu_3 &= (10, 10, 10, 10, 10, 10)' & \Psi_3 &= \text{diag}(0.2, 0.2, 0.2, 0.2, 0.2, 0.2) \end{aligned}$$

$$\Lambda_1 = \begin{pmatrix} 0.50 & 1.00 \\ 1.00 & 0.45 \\ 0.05 & -0.50 \\ -0.60 & 0.50 \\ 0.50 & 0.10 \\ 1.00 & -0.15 \end{pmatrix} \quad \Lambda_2 = \begin{pmatrix} 0.10 & 0.20 \\ 0.20 & 0.50 \\ 1.00 & -1.00 \\ -0.20 & 0.50 \\ 1.00 & 0.70 \\ 1.20 & -0.30 \end{pmatrix} \quad \Lambda_3 = \begin{pmatrix} 0.10 & 0.20 \\ 0.20 & 0.00 \\ 1.00 & 0.00 \\ -0.20 & 0.00 \\ 1.00 & 0.00 \\ 0.00 & -1.30 \end{pmatrix}.$$

Figure 4.1 shows a specimen of randomly generated data from the given mixture.

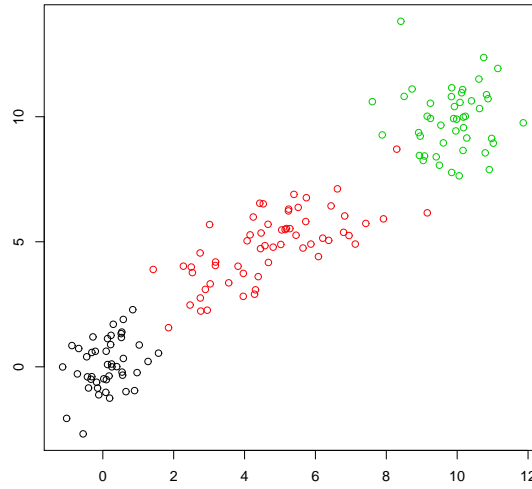


Figure 4.1: A specimen of 150 data points generated from the mixture (the first two coordinates are plotted, groups in black, red and green)

Our analysis begins by running the AECM algorithm on the generated sample, and considering the following six settings, namely:

- S1. a "virtually" unconstrained approach (i.e. $c_{\text{noise}} = c_{\text{load}} = 10^{10}$) without trimming ($\alpha = 0$),
- S2. an adequate constraint on Ψ_g , no constraint on Λ_g ($c_{\text{noise}} = 5, c_{\text{load}} = 10^{10}$) and no trimming ($\alpha = 0$),
- S3. adequate constraints on Ψ_g and Λ_g ($c_{\text{noise}} = 5, c_{\text{load}} = 3$), and still no trimming ($\alpha = 0$),
- S4. a "virtually" unconstrained approach (i.e. $c_{\text{noise}} = c_{\text{load}} = 10^{10}$) with trimming ($\alpha = 0.06$),
- S5. an adequate constraint on Ψ_g , no constraint on Λ_g ($c_{\text{noise}} = 5, c_{\text{load}} = 10^{10}$), with trimming ($\alpha = 0.06$),
- S6. adequate constraints on Ψ_g and Λ_g ($c_{\text{noise}} = 5, c_{\text{load}} = 3$), with trimming ($\alpha = 0.06$)

It is worth noticing that when setting $c_{\text{noise}} = 10^{10}$ we want to discard singularities, and allow the estimation to move in a wide parameter space that contains the global maximum, among several local ones. In this situation the estimation could incur into spurious solutions. We expect

that the algorithm improve its performances when giving the "right" constraints. The adequate constraints can be evaluated by obtaining the maximum ratio among the eigenvalues of Ψ_g and among the singular values of Λ_g . We have that the singular values of Λ_1 are (3.069, 1.528), of Λ_2 are (3.777, 1.873) and of Λ_3 are (2.091, 1.729) hence we derive $c_{load} \geq 2.471$; while the diagonal elements of Ψ_g are 0.1, 0.4, and 0.2 so that $c_{noise} \geq 4$. We applied also trimming to the artificially generated data, to see the effect of an unneeded elimination of the outermost points in the model estimation and subsequent classification. We evaluate the performance of the algorithm by calculating the average misclassification error η , over 100 repetitions of the estimation procedure. The misclassification error is defined as the relative frequency of points of the sample erroneously labeled, taking into account that noise and pointwise contamination (when added) should be identified, as they virtually do not belong to the three groups. We see that the algorithm, applied without trimming, give a perfect classification with and without constraints, due to fact that estimation is performed along 10 random initializations. While adding trimming, the misclassification error is almost equal to the trimming level (as expected). The results are summarized in Table 4.1. Moreover, we observed that the other parameters, such as the means μ_g , and Ψ_g , Λ_g for $g = 1, 2, 3$ are close to the values from which the data have been generated.

Table 4.1: Misclassification error η (average on 100 repetitions of the estimation procedure) of the AECM algorithm with settings S1-S6, applied on the artificially generated data

	S1	S2	S3	S4	S5	S6
c_{noise}	10^{10}	5	5	10^{10}	5	5
c_{load}	10^{10}	10^{10}	3	10^{10}	10^{10}	3
α	0	0	0	0.06	0.06	0.06
η	0.33%	0.04%	0.00%	6.45%	6.13%	6.00%

Afterwards, we have considered 3 different scenarios.

D+N: 10 points of uniform noise have been added around the data,

D+PC: 10 points of pointwise contamination have been added outside the range of the data,

D+N+PC: both the uniform noise and the pointwise contamination have been added to the data.

We applied the algorithm to the different datasets in the six previous settings S1-S6 (i.e. with/without constraints and trimming) and we calculated the misclassification error. Results in the first row of Table 4.2 show that trimming is very effective to identify and discard noise in the data, and constraints contribute slightly, to reach perfect classification. The misclassification error (re-

ported in the second row of Table 4.2) shows that we need trimming *and* constraints to achieve a very good behavior of the algorithm. Noise and pointwise contamination could cause very messy estimation, as it is seen in the first three columns of the Table, where we only rely/do not rely on constraints. Further, we observe that trimming is a good strategy when dealing with uniform noise, but it is not able to resist pointwise contamination. If we want to be protected against all type of data contamination we do need both the use of constrained estimation and trimming.

Table 4.2: Misclassification error η (average on 100 repetitions of the estimation procedure) of the AECM algorithm with settings S1-S6, applied on different data sets

	S1	S2	S3	S4	S5	S6
c_{noise}	10^{10}	5	5	10^{10}	5	5
c_{load}	10^{10}	10^{10}	3	10^{10}	10^{10}	3
α	0	0	0	0.06	0.06	0.06
D+N	0.3348	0.4856	0.5357	0.0305	0.0033	0.0000
D+PC	0.2811	0.2659	0.2837	0.0465	0.0071	0.0031
D+N+PC	0.4035	0.5299	0.5294	0.0918	0.0124	0.0064

4.1.1 Properties of the estimators for the mixture parameters

Now, we want to perform a second analysis on this artificial data and our main interest here is in assessing the effect of trimming and constraints on the properties of the model estimators. Namely, we will estimate their bias and mean square error when the data is affected by noise and/or pointwise contamination. We will consider the same four scenarios we considered before, i.e.:

D: the artificially generated data,

D+N: the data with added noise,

D+PC: the data with added pointwise contamination,

D+N+PC: the data with added noise and pointwise contamination.

We apply in the four scenarios the algorithm for estimating a trimmed MFA model, exploring the six settings on c_{noise} , c_{load} and α that have been shown in Table 4.2. For sake of space, we report our results only on the more interesting cases.

The benchmark of all simulations is given by the results that we obtain on artificial data drawn from a given MFA without outliers, and they are shown in the first column of Table 4.3.

In each experiment, we draw 1000 times a sample of size $n = 150$ from the mixture described at the beginning of this Section, and we estimate the model parameters for the trimmed MFA using the algorithm presented in the previous Section 3.2. We set $c_{noise} = c_{load} = 10^{10}$ and $\alpha = 0$ for this first case, as no outliers are added to the samples.

Notice that the considered estimators in each component are vectors (apart from π_g which are scalar quantities, for $g = 1, \dots, G$). We are interested in providing synthetic measures of their properties, such as bias and mean square error (MSE). As usual, let \hat{T} be an estimator for the scalar parameter t , then the bias of \hat{T} is given by $bias(\hat{T}) = \mathbb{E}(\hat{T}) - t$, i.e. it is the signed absolute deviation of the expected value $\mathbb{E}(\hat{T})$ from t . Therefore, we would have 6 biases for each component of the mean μ_g , 6 for $diag(\Psi_g)$ and 12 for Λ_g . On the other side, MSE is defined as a scalar quantity, namely $\mathbb{E}(|\hat{T} - t|^2) = \text{trace}(\text{Var}(\hat{T})) + bias(\hat{T})^2$, also for vector estimators. Hence, we adopted a synthesis of each parameter biases by considering the mean of their absolute values on each component. Below the bias, in Table 4.3, we provide the MSE in parenthesis.

Then, the second experiment consists in drawing 1000 samples as before, and adding 10 points of random uniform noise to each of them; the bias and mean square error for the model estimators increase dramatically, with $c_{noise} = c_{load} = 10^{10}$ and $\alpha = 0$ (results are displayed in second column of Table 4.3). On the other hand, results go back to the same order of magnitude as the benchmark if we impose $c_{noise} = 5$, $c_{load} = 3$ and $\alpha = 0.06$, as it is shown in the second column of Table 4.4.

The third experiment is based on 1000 samples, with 10 points of pointwise contamination randomly added. We observed a huge increase of the bias and mean square error for the model estimators, without appropriate constraints and level of trimming (see results in third column of Table 4.3), but whenever we run the algorithm with $c_{noise} = 5$, $c_{load} = 3$ and $\alpha = 0.06$ we came back to results very close to the benchmark, shown in the third column of Table 4.4.

The fourth experiment has been developed by considering added random noise and pointwise contamination to the 1000 drawn samples. The results on bias and mean square error for the case of estimating the trimmed MFA with $c_{noise} = c_{load} = 10^{10}$ and $\alpha = 0$, in the fourth column of Table 4.3, show the harmful effects of distorted inference. On the other side, when we applied reasonable constraints $c_{noise} = 5$, $c_{load} = 3$ and a trimming level $\alpha = 0.12$ to cope with the added outliers, we got the results in the fourth column of Table 4.4. We see that robust inference allows reduced bias and mean square error, even in case of both sparse outliers and concentrated leverage points.

Finally the scheme of simulations on the four data sets, in the six estimation settings, have been repeated considering a triple sample size ($n = 450$). All results are summarized in Table 4.3 when $c_{noise} = c_{load} = 10^{10}$ and $\alpha = 0$ to be compared with results in Table 4.4 in which appropriate constraints and trimming have been used along the estimation, to see the improved properties of the estimators when the sample size increases.

Table 4.3: Case without trimming and constraints.

Bias as the sum of absolute deviations, followed by MSE (in parentheses) of the parameter estimators $\hat{\pi}_i, \hat{\mu}_i, \hat{\Psi}_i, \hat{\Lambda}_i$, for $i = 1, 2, 3$ when dealing with different datasets. The column labels refer to the different scenarios, namely D stays for *data*, D+N stays for *data and added noise*, D+PC stays for *data and pointwise contamination*, D+N+PC stays for *data with added noise and pointwise contamination*.

	D	D+N	D+PC	D+N+PC	3*(D+N+PC)
$\hat{\tau}_1$	0.0013 (0)	0.124 (0.0166)	0.1219 (0.0153)	0.1893 (0.038)	0.1649 (0.0312)
$\hat{\tau}_2$	0.0065 (0)	0.0877 (0.0089)	0.1359 (0.0189)	0.324 (0.1072)	0.2097 (0.048)
$\hat{\tau}_3$	0.0053 (0)	0.2118 (0.0461)	0.2579 (0.067)	0.1347 (0.0204)	0.0448 (0.006)
$\hat{\mu}_1$	0.018 (0.305)	1.506 (31.736)	1.955 (39.371)	11.534 (659.863)	13.093 (1140.549)
$\hat{\mu}_2$	0.006 (0.497)	5.15 (345.478)	5.87 (133.962)	1.845 (99.898)	3.728 (207.869)
$\hat{\mu}_3$	0.059 (0.712)	11.651 (729.452)	2.63 (17.135)	11.949 (867.905)	8.159 (790.799)
$\hat{\Psi}_1$	0.013 (0.027)	0.435 (11.373)	0.043 (0.054)	0.564 (115.608)	1.274 (272.187)
$\hat{\Psi}_2$	0.066 (0.172)	4.622 (1520.318)	0.203 (0.79)	3.919 (651.607)	9.339 (1819.385)
$\hat{\Psi}_3$	0.239 (2.039)	15.986 (5634.188)	0.38 (1.891)	14.736 (4557.463)	25.429 (10196.275)
$\hat{\Lambda}_1$	0.534 (82.817)	0.534 (82.817)	0.534 (96.75)	0.498 (300.57)	0.522 (361.565)
$\hat{\Lambda}_2$	0.608 (172.601)	0.608 (172.601)	0.551 (86.747)	0.642 (304.718)	0.653 (633.379)
$\hat{\Lambda}_3$	0.335 (404.326)	0.335 (404.326)	0.354 (53.063)	0.373 (284.401)	0.341 (326.672)

The distributions of the estimators for the model parameters can be represented through some box plots, and some of them are shown in Figure 4.2, namely with reference to $\hat{\pi}_3$ (upper panel), $\hat{\mu}_3[1, 1]$ (second panel), $\hat{\Psi}_3[1, 1]$ (third panel) and $\hat{\Lambda}_3[1, 1]$ (bottom panel). We can see,

Table 4.4: Case with trimming and constraints.

Bias as the sum of absolute deviations, followed by MSE (in parentheses) of the parameter estimators $\hat{\pi}_i, \hat{\mu}_i, \hat{\Psi}_i, \hat{\Lambda}_i$, for $i = 1, 2, 3$ when dealing with different datasets. The column labels refer to the different scenarios, namely D stays for *data*, D+N stays for *data and added noise*, D+PC stays for *data and pointwise contamination*, D+N+PC stays for *data with added noise and pointwise contamination*.

	D	D+N	D+PC	D+N+PC	3*(D+N+PC)
$\hat{\tau}_1$	0.0151 (0.0002)	0.0002 (0)	0.0011 (0)	0 (0)	0.0006 (0)
$\hat{\tau}_2$	0.0195 (0.0004)	0.0014 (0)	0.0011 (0)	0.0034 (0)	0.0006 (0)
$\hat{\tau}_3$	0.0044 (0)	0.0012 (0)	0.0001 (0)	0.0034 (0)	0.0001 (0)
$\hat{\mu}_1$	0.006 (0.117)	0.010 (0.159)	0.017 (0.215)	0.006 (0.226)	0.001 (0.038)
$\hat{\mu}_2$	0.006 (0.190)	0.004 (0.219)	0.002 (0.165)	0.007 (0.703)	0.009 (0.158)
$\hat{\mu}_3$	0.008 (0.177)	0.020 (0.302)	0.004 (0.154)	0.062 (0.787)	0.018 (0.231)
$\hat{\Psi}_1$	0.013 (0.010)	0.029 (0.025)	0.026 (0.024)	0.032 (0.035)	0.022 (0.009)
$\hat{\Psi}_2$	0.066 (0.089)	0.044 (0.082)	0.045 (0.081)	0.046 (0.102)	0.042 (0.058)
$\hat{\Psi}_3$	0.066 (0.108)	0.072 (0.158)	0.075 (0.154)	0.076 (0.178)	0.075 (0.158)
$\hat{\Lambda}_1$	0.516 (13.168)	0.512 (14.828)	0.516 (12.992)	0.546 (14.711)	0.524 (13.674)
$\hat{\Lambda}_2$	0.568 (11.049)	0.569 (12.311)	0.569 (12.036)	0.586 (13.875)	0.578 (12.832)
$\hat{\Lambda}_3$	0.330 (9.377)	0.353 (11.164)	0.354 (12.259)	0.341 (13.069)	0.342 (13.064)

in a direct comparison, that the estimation algorithm with adequate trimming and constraints is able to resist all type of outlying data. In each panel, the first boxplot on the left provides the benchmark of the following five ones, as it shows the distribution of the estimator when the data has been drawn from the mixture. The second boxplot (from left to right) in each panel shows the distribution of the estimator when we employ constraints and trimming along the estimation on data and added uniform noise. The third boxplot refers to the case in which we deal with data and pointwise contamination, and the good results are obtained because we are employing constraints and trimming. The fourth box plot has been obtained when considering both noise and pointwise contamination, and robust estimation. The fifth box plot shows the

effects of noise and pointwise contamination when the estimation procedure does not employ constraints and trimming. Finally the sixth box plot in all panels reports the case of robust estimation performed on a triple sample size, still with noise and pointwise contamination.

4.2 Real data: the AIS data set

As an illustration, we apply the proposed technique to the Australian Institute of Sports (AIS) data, which is a famous benchmark dataset in the multivariate literature, originally reported by Cook and Weisberg (1994) and subsequently analyzed by Azzalini and Dalla Valle (1996), among many other authors. The dataset consists of $p = 11$ physical and hematological measurements on 202 athletes (100 females and 102 males) in different sports, and is available within the R package *sn* (Azzalini, 2011). The observed variables are: red cell count (RCC), white cell count (WCC), Hematocrit (Hc), Hemoglobin (Hg), plasma ferritin concentration (Fe), body mass index, weight/height² (BMI), sum of skin folds (SSF), body fat percentage (Bfat), lean body mass (LBM), height, cm (Ht), weight, kg (Wt), a part from Sex and kind of Sport. A partial scatterplot of the AIS dataset is given in Figure 4.3, and Table 4.5 provides summary information.

Table 4.5: Summary information for the AIS dataset

	female athletes						male athletes					
	min	Q1	Me	Q3	max	mean	min	Q1	Me	Q3	max	mean
<i>RCC</i>	3.8	4.2	4.4	4.5	5.3	4.4	4.1	4.9	5.0	5.2	6.7	5.0
<i>WCC</i>	3.3	5.8	6.7	8.0	13.3	7.0	3.9	6.0	7.1	8.4	14.3	7.2
<i>Hc</i>	35.9	38.3	40.6	42.3	47.1	40.5	40.3	44.2	45.5	46.8	59.7	45.6
<i>Hg</i>	11.6	12.7	13.5	14.3	15.9	13.6	13.5	14.9	15.5	15.9	19.2	15.6
<i>Fe</i>	12.0	36.0	50.0	71.5	182.0	57.0	8.0	55.0	89.5	123.5	234.0	96.4
<i>BMI</i>	16.8	20.3	21.8	23.4	31.9	22.0	19.6	22.3	23.6	25.2	34.4	23.9
<i>SSF</i>	33.8	59.3	81.8	107.4	200.8	87.0	28.0	37.5	47.7	58.1	113.5	51.4
<i>Bfat</i>	8.1	13.2	17.9	21.4	35.5	17.8	5.6	7.0	8.6	10.0	19.9	9.3
<i>LBM</i>	34.4	51.9	54.9	59.4	73.0	54.9	48.0	68.0	74.5	80.8	106.0	74.7
<i>Ht</i>	148.9	171.0	175.0	179.7	195.9	174.6	165.3	179.7	185.6	191.0	209.4	185.5
<i>Wt</i>	37.8	60.1	68.1	74.4	96.3	67.3	53.8	73.9	83.0	90.3	123.2	82.5

Our purpose is to provide a model for the entire dataset, and classify athletes by Sex. Let us begin our analysis by fitting a mixture of multivariate Gaussian distributions, using *Mclust*

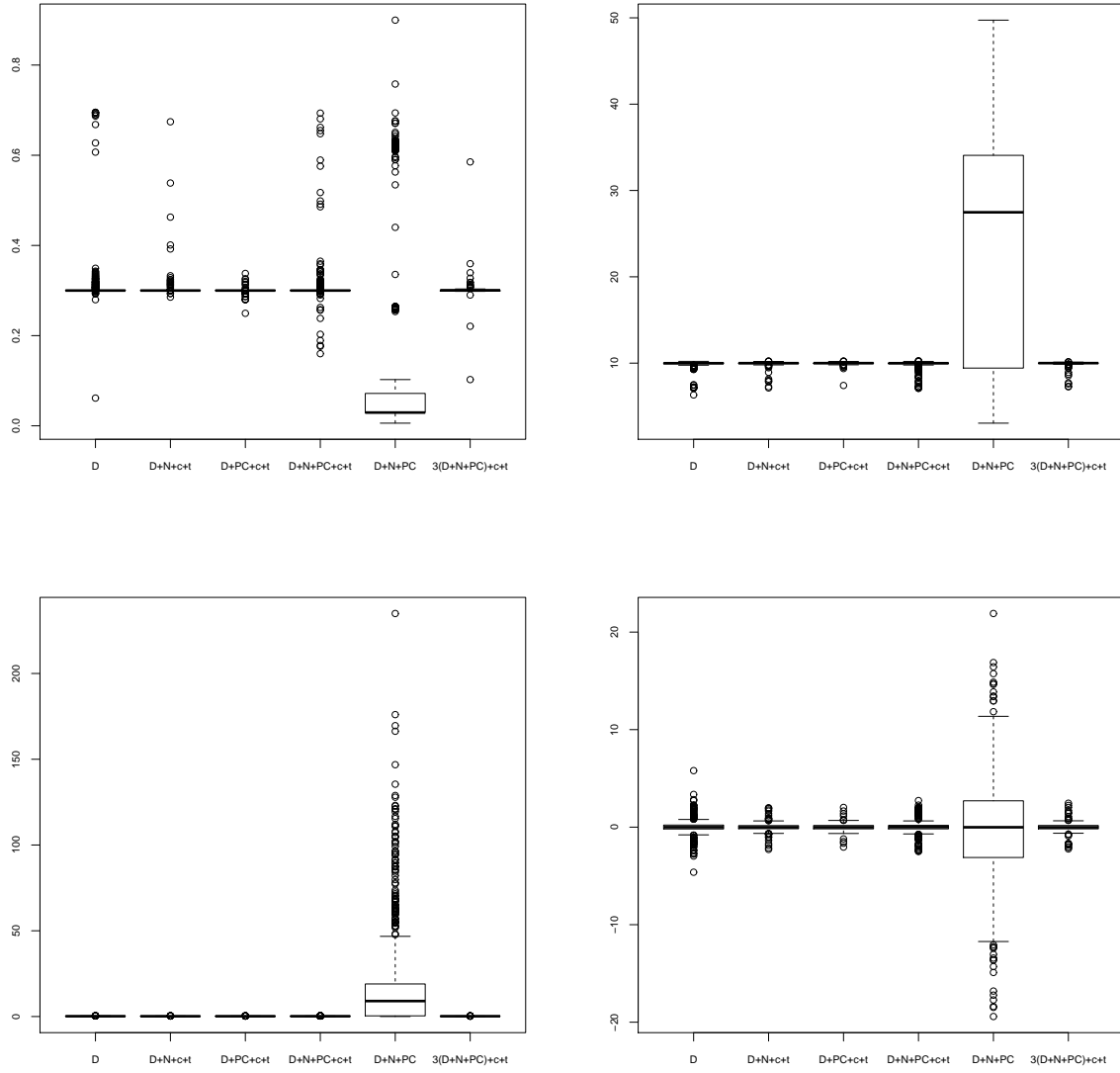


Figure 4.2: Boxplots of some of the simulated distributions of the mixture parameters:

the simulated distribution of $\hat{\pi}_3$, estimator for $\pi_3 = 0.3$ (upper panel);

the simulated distribution of $\hat{\mu}_3[1]$, estimator for $\mu_3[1] = 10$ (second panel from above);

the simulated distribution of $\hat{\Psi}_3[1,1]$, estimator for $\Psi_3[1,1] = 0.2$ (third panel from above); and

the simulated distribution of $\hat{\Lambda}_3[1,1]$, estimator for $\Lambda_3[1,1] = 0.1$ (fourth panel).

The labels on the horizontal axis refers to the six settings, namely “D” stays for *data*, “D+N” stays for *data and added noise*, “D+N+PC” stays for *data with added noise and pointwise contamination*, while “c + t” has been added to denote the cases in which the estimation has been performed *using constraints and trimming*.

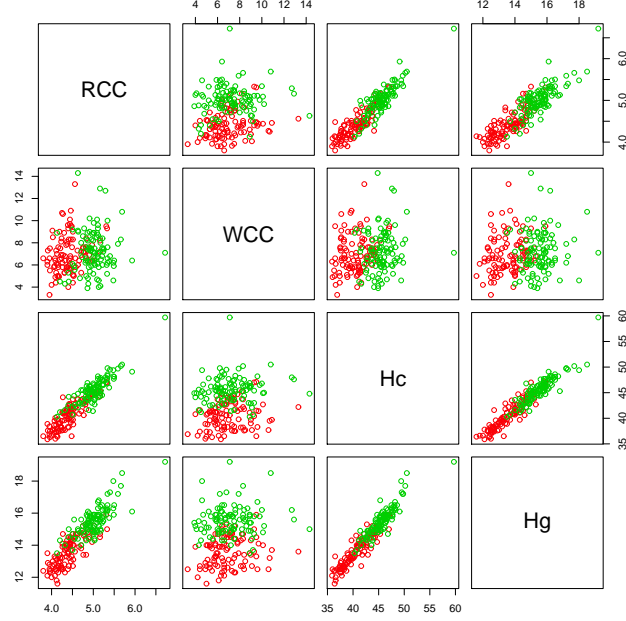


Figure 4.3: Scatterplot of some pairs of the AIS variables (female data in red, male in blue)

package in R. The routine *mclustBIC* fits a set of normal mixture models, on the base of the parameters you set in its call. We considered from 1 to 9 components in the mixture and different patterns for the covariance matrices, from the more constrained homoscedastic model, to the more general heteroscedastic one. After the estimation, *mclustBIC* provides a model selection procedure based on the BIC, a well-known penalized likelihood criterium. In Figure 4.4 the BIC values for each kind of model are shown, and for each choice of the number of mixture components. The three letters in the acronym of the models stand respectively for the *volume*, the *shape* and *orientation* of the ellipsoids of equal probability of the components, which could be Equal (hence E) or Variable (V) across the components. Notice that the shape may also be Isotropic (hence the letter I denotes spherical ellipsoids), in this case also the orientation is the same. Hence, we see that *Mclust* suggests an EEV model (ellipsoidal, equal volume and shape, different orientation of the component scatters) with $G = 2$ components, providing the highest BIC value, i.e. $BIC = -10251.6$. Now, if we employ the model suggested by *Mclust* to classify AIS data, we obtain 18 misclassified units, i.e., a misclassification error equal to $18/202 = 9.4\%$. In Figure 4.4 we can see the classification results. Surely this is not an easy dataset for classification, due to the apparent class overlapping we saw in the first scatterplot in Figure 4.3.

To improve the classification, we may exploit the conjecture that a strong correlation ex-

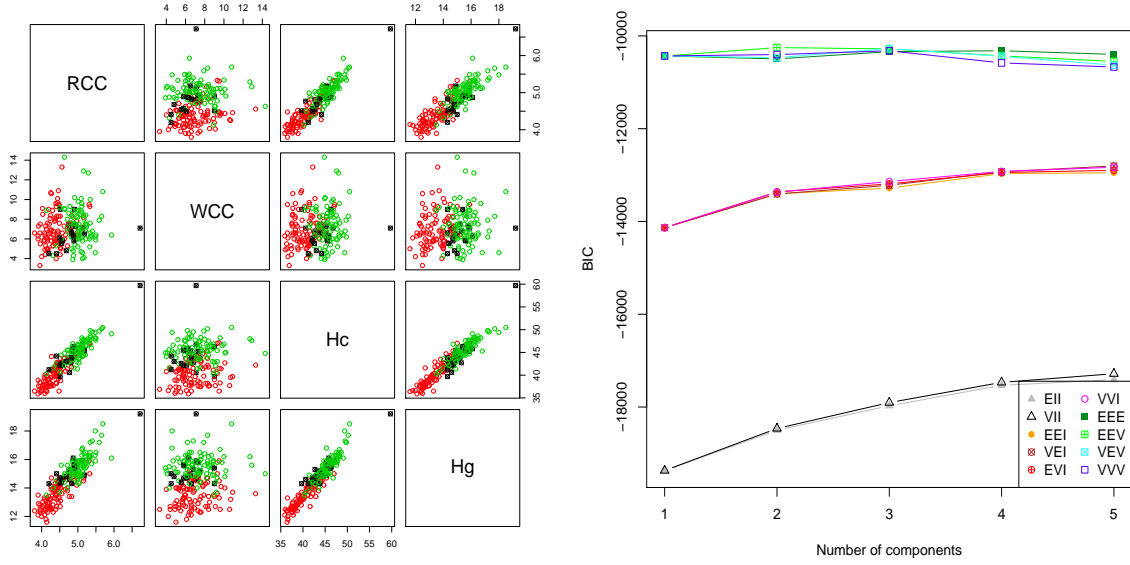


Figure 4.4: The classification of AIS data obtained through the best model from Mclust (left panel, with female data in red, male in blue, misclassified units as black circlecrosses), and the graphical tool for model selection (right panel)

ists among the hematological and physical measurement. Therefore we fit mixtures of factor analyzers, assuming the existence of some underlying unknown factors (like nutritional status, hematological composition, overweight status indices, and so on) which jointly explains the observed measurements. Through the underlying factors, we aim at finding a perspective on data which disentangle the overlapping components. To avoid variables having a greater impact in the model (which is not affine equivariant) due to different scales, before performing the estimation, the variables have been standardized to have zero mean and unit standard deviation. We begin by adopting the *pGmm* package from R, that fits mixtures of factors analyzers with patterned covariances. Parsimonious Gaussian mixtures are obtained by constraining the loading Λ_j and the errors Ψ_j to be equal or not among the components. We employed the routine *pGmmEM*, considering from 1 to 9 components, and number of underlying factors d ranging from 1 to 6, with 30 different random initialization, to provide the best iteration (in terms of BIC) for each case. The best model is a CUU mixture model with $d = 4$ factors and $G = 3$ components, with $BIC = -3127.424$. CUU means Constrained loading matrices $\Lambda_j = \Lambda$ and Unconstrained error matrices $\Psi_g = \omega_g \Delta_g$, where Δ_g are normalized diagonal matrices and ω_g is a real value varying across components. Using this model to classify athletes, we got 109 misclassified units and we discarded it.

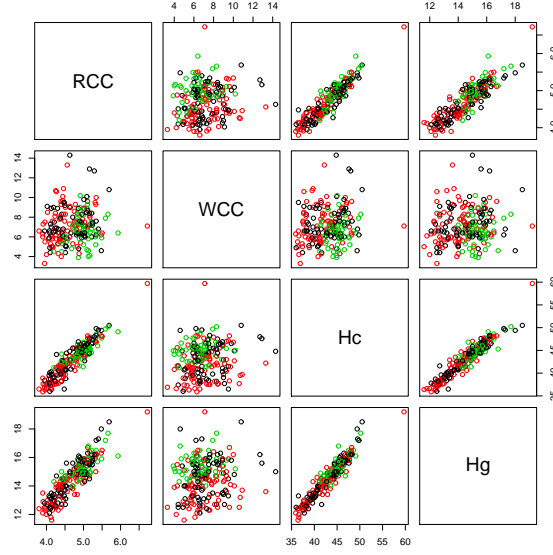


Figure 4.5: The classification of AIS data obtained through the best UUU model from $pGmm$ with $G = 2$ and $d = 4$ (female data in red, male in green, misclassified units in black)

As a second attempt using $pGmm$, we estimated a UUU model by setting $G = 2$ components, and $d = 6$. The acronym UUU means that we leave the estimation of loadings Λ_j and errors Ψ_j unconstrained. Based on 30 random starts, the best UUU model had $BIC = -3330.306$, and the consequent classification of the AIS dataset produces 72 misclassified units (misclassification error=35.6%), that are visualized in Figure 4.5.

Finally, we want to show the performances of our trimmed and constrained estimation for MFA on the AIS data. All the results are generated by the procedure described in Section 3.2, are based on 30 random initializations and returning the best obtained solution of the parameters, in terms of highest value of the final likelihood.

Table 4.6: Trimmed and constrained MFA estimation on the AIS data set (best results over 30 random initializations). Misclassification error η (in percentage) under different settings

c_{noise}	10^{10}	45	10^{10}	45	10^{10}	45	10^{10}	45
c_{load}	10^{10}	10^{10}	10	10	10^{10}	10^{10}	10	10
α	0	0	0	0	0.05	0.05	0.05	0.05
η	0.0891	0.1881	0.4554	0.0842	0.1039	0.1782	0.4505	0.0149

We see that the best solution, with only 3 misclassified points, has been obtained by combining trimming ($\alpha = 0.05$) and constrained estimation of Ψ_g ($c_{noise} = 45$) and Λ_g ($c_{load} = 10$),

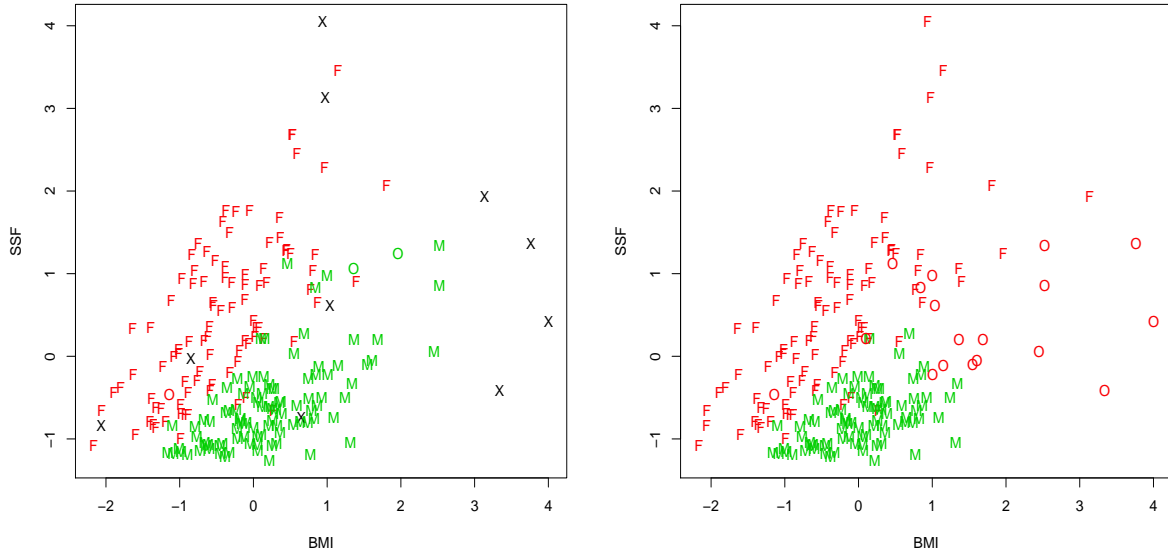


Figure 4.6: Classification of AIS data with fitted trimmed and constrained MFA (left panel), compared to non-robust MFA, i.e. the fitted model in 1st row of Table 4.6 (right panel). Misclassified data are denoted by O, trimmed data by X.

with $d = 6$. The choice of $d = 6$ has been motivated by performing a factor analysis on the observations coming from the group of male athletes, in which we may employ the screw plot and test the hypothesis that 6 factors are sufficient, with chi square statistic equal to 97.81 on 4 degrees of freedom, and $p\text{-value} = 2.88e - 20$. In any case, results with $d \neq 6$ have also been checked. The constraints, and in first place the constraint on Ψ_g , play an important role (compare results in columns 2-4-6 and 8 to the ones displayed in the odd columns), but trimming is needed to reach the best result. This is motivated by the fact that the data, as a whole, are not following a 11-dimension multivariate Gaussian, as it can be easily checked by performing a previous Mardia test. Two results of the fitted models and the subsequent classifications are displayed in Figure 4.6, by selecting the 2 variables in the scatterplot that allow a better vision of trimmed and misclassified units. We have chosen to represent the best solution (upper panel), with only 3 misclassified points, denoted by “O” in the graph, and with 10 trimmed points, denoted by “X”. In the lower panel, to make a comparison, we report classification results obtained by the fitted model in first row of Table 4.6. In this second case, we were doing an almost unconstrained estimation of Ψ_g and Λ_g and we were not applying trimming, obtaining 18 misclassified observations.

The misclassified observations are in rows 70, 73, and 121 in the AIS dataset. Two misclas-

sified units are among female athletes, one is among male athletes. The density of the mixture components for the observation in position 73 are close (0.000021 and 0.00034), while for the other two observations they are neatly different.

Finally, we recall that trimmed observations were discarded to provide robustness to the parameter estimation. After estimating the model, hence, it makes sense to classify also these observations. The trimmed observations are in rows 11, 56, 75, 93, 99, 133, 160, 163, 166, 178, and if we assign them by the Bayes rule to the component g having greater value of $D_g(\mathbf{x}; \theta) = \phi_p(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g) \pi_g$, we classify the first five in the female group of athletes, and the second group of five in the male group. This means that all the trimmed observations have been assigned to their true group. Table 4.7 shows the details of the classification, and Figure 4.7 plots the final result of the robust model fitting.

Table 4.7: Trimmed units in the AIS data set and their final classification

unit	$D_1(\mathbf{x}; \theta) = \phi_p(\mathbf{x}; \mu_1, \boldsymbol{\Lambda}_1 \boldsymbol{\Lambda}_1' + \boldsymbol{\Psi}_1) \pi_1$	$D_2(\mathbf{x}; \theta) = \phi_p(\mathbf{x}; \mu_2, \boldsymbol{\Lambda}_2 \boldsymbol{\Lambda}_2' + \boldsymbol{\Psi}_2) \pi_2$	Sex
11	1.4 e-15	9.2 e-20	F
56	7.2 e-08	4.5 e-25	F
75	5.2 e-09	1.2 e-11	F
93	1.7 e-07	1.0 e-10	F
99	1.2 e-09	6.4 e-70	F
133	9.8 e-85	3.2 e-12	M
160	9.9 e-74	1.5 e-08	M
163	9.9 e-87	2.0 e-08	M
166	2.2 e-16	1.4 e-13	M
178	3.1 e-23	3.8 e-13	M

As a last analysis on AIS dataset, we are interested in factor interpretation.

The rotated factor loading matrices have been obtained by employing a Gradient Projection algorithm, available through the R package *GPArotation* (Bernaards and Jennrich, 2005; Browne, 2001). We opted for an oblimin transformation, which yielded results shown in Table 4.8. We observe that the two groups highlight the same factors, while in a slightly different order of importance. The first factor for the group of observations for female athletes, may be labelled as a *hematological factor*, with a very high loading on *Hc*, followed by *RCC* and *Hg*. The second factor, loading heavily on *Ht*, and in a lesser extent on *Wt* and *LBM*, may be denoted as a *general nutritional status*. The third and the fourth factors are related only to *Fe* and

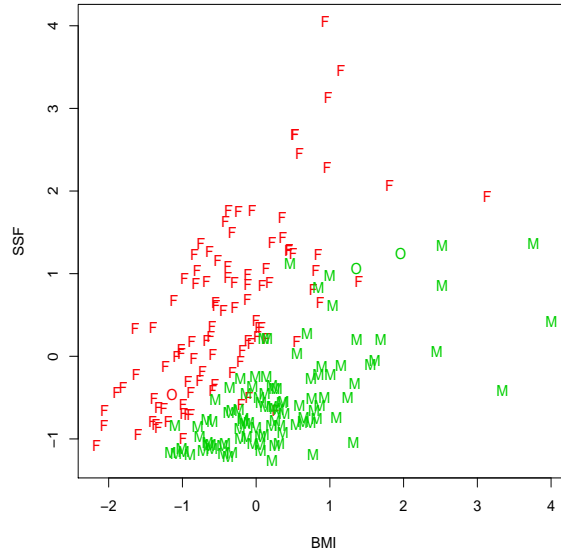


Figure 4.7: Classification of AIS data after classifying also trimmed observations. The three misclassified points are denoted by O, and they represent only 1.5% of the data.

BMI , respectively. The fifth factor can be viewed as a *overweight assessment index* since SSF and $Bfat$ load highly on it. The sixth factor is related only to WCC . Noticing that WCC is not joined into the *hematological factor*, we observe that the specific role of lymphocytes, cells of the immune system that are involved in defending the body against both infectious disease and foreign invaders, seems to be pointed out. Analogous comments may be done on the factor loadings for the group of male athletes.

5 Concluding remarks

In this paper we propose a robust estimation for the mixture of Gaussian factors model. To resist pointwise contamination and sparse outliers that could arise in data collection, we adopt and incorporate a trimming procedure along the iterations of the EM algorithm. The key idea is that a small portion of observations which are highly unlikely to occur, under the current fitted model assumption, are discarded from contributing to the parameter estimates. Furthermore, to reduce spurious solutions and avoid singularities of the likelihood, we implement a constrained ML estimation for the component covariances. Results from Monte Carlo experiments show that bias and MSE of the estimators in several cases of contaminated data are comparable to results obtained on data without noise. Finally, the analysis on a real dataset illustrates that

Table 4.8: Factor loadings in the AIS data set

rotated Λ_1 (female athletes)						
<i>RCC</i>	0.697	-0.006	-0.009	-0.055	0.001	-0.035
<i>WCC</i>	0.000	0.009	0.000	-0.015	0.012	-0.941
<i>Hc</i>	0.794	-0.015	0.040	-0.004	0.010	0.026
<i>Hg</i>	0.682	0.021	-0.025	0.047	-0.002	0.007
<i>Fe</i>	0.002	-0.005	-0.510	0.003	-0.004	0.000
<i>BMI</i>	0.029	-0.008	0.023	0.644	-0.316	-0.057
<i>SSF</i>	-0.040	-0.012	-0.037	0.033	-0.889	-0.017
<i>Bfat</i>	0.014	-0.024	0.013	-0.007	-0.826	0.008
<i>LBM</i>	0.022	-0.419	0.020	0.295	0.054	-0.025
<i>Ht</i>	0.020	-0.924	0.023	-0.128	-0.076	-0.002
<i>Wt</i>	0.029	-0.468	0.023	0.330	-0.235	-0.031
rotated Λ_2 (male athletes)						
<i>RCC</i>	-0.033	0.663	0.077	-0.015	-0.025	-0.053
<i>WCC</i>	0.003	-0.004	0.024	-0.003	1.024	0.013
<i>Hc</i>	0.048	0.622	-0.008	0.005	0.036	-0.048
<i>Hg</i>	-0.002	0.604	-0.051	0.009	0.001	0.079
<i>Fe</i>	0.010	-0.006	0.027	1.103	-0.004	0.008
<i>BMI</i>	-0.371	0.109	-0.656	0.074	0.070	-0.261
<i>SSF</i>	-0.616	0.002	0.009	-0.001	0.015	-0.026
<i>Bfat</i>	-0.610	-0.009	0.003	0.007	-0.000	0.040
<i>LBM</i>	0.036	0.071	-0.344	0.037	0.053	-0.885
<i>Ht</i>	0.036	0.005	0.170	-0.022	0.009	-1.157
<i>Wt</i>	-0.222	0.071	-0.357	0.042	0.056	-0.884

robust estimation leads to better classification and provides direct interpretation of the factor loadings.

Further investigations are needed to tune the choice of the parameters, such as the portion of trimming data and the values of the constraints. Though interesting, this issue is beyond the scope of the present paper. Surely, the researcher may specify the partial information he may have about the shape of the expected clusters from the data at hand, hence providing a part of these parameters. Then, data-dependent diagnostic based on trimmed BIC notions (Neykov et al., 2007) and/or graphical tools such as the ones in García-Escudero et al. (2011),

conveniently adapted to the specific case, could assist in taking appropriate choices for the rest of the parameters. The encouraging results here obtained suggest a deeper discussion of these implementation details in a future work.

Acknowledgements

This research is partially supported by the Spanish Ministerio de Ciencia e Innovación, grant MTM2011-28657-C02-01, by Consejería de Educación de la Junta de Castilla y León, grant VA212U13, and by grant FAR 2013 from the University of Milano-Bicocca.

References

- Azzalini A., Dalla Valle A., The multivariate skew-normal distribution, *Biometrika*, 83 (4), 715–726, (1996).
- Baek J., McLachlan G., Flack L., Mixtures of Factor Analyzers with Common Factor Loadings: Applications to the Clustering and Visualization of High-Dimensional Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (7), 1298–1309, (2010).
- Baek J., McLachlan G., Mixtures of common t-factor analyzers for clustering high-dimensional microarray data, *Bioinformatics*, 27 (9), 1269–1276, (2011).
- Bernaards C. A., Jennrich R. I., Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis, *Educational and Psychological Measurement*, 65 (5), 676–696, (2005).
- Bickel D.R., Robust cluster analysis of microarray gene expression data with the number of clusters determined biologically, *Bioinformatics*, 19 (7), 818–824, (2003).
- Bishop C. M., Tipping M. E., A Hierarchical Latent Variable Model for Data Visualization, *IEEE Transactions on Pattern analysis and Machine Intelligence*, 20, 281–293, (1998).
- Browne M. W., An overview of analytic rotation in exploratory factor analysis, *Multivariate Behavioral Research*, 36 (1), 111–150, (2001).
- Campbell J., Fraley C., Murtagh F., Raftery A., Linear flaw detection in woven textiles using model-based clustering, *Pattern Recognition Letters*, 18 (14), 1539–1548, (1997).
- Cook R. D., Weisberg S., *An introduction to regression graphics*, vol. 405, John Wiley & Sons, 1994.
- Coretto P., Hennig C., Maximum likelihood estimation of heterogeneous mixtures of Gaussian and uniform distributions, *Journal of Statistical Planning and Inference*, 141 (1) 462–473, (2011).
- Fokoué E., Titterton D., Mixtures of factor analysers. Bayesian estimation and inference by stochastic simulation, *Machine Learning*, 50 (1-2), 73–94, (2003).

- Fraley C., Raftery A. E., How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis, *Computer Journal*, 41 (8), 578–588, (1998).
- Fritz H., García-Escudero L., Mayo-Iscar A., A fast algorithm for robust constrained clustering, *Computational Statistics & Data Analysis*, (61), 124–136, (2013).
- Gallegos M., Ritter G., Trimmed ML estimation of contaminated mixtures, *Sankhya (Ser. A)*, (71), 164–220, (2009).
- García-Escudero L. A., Gordaliza A., Matrán C., Mayo-Iscar A., A General Trimming Approach to Robust Cluster Analysis, *The Annals of Statistics*, 36 (3) 1324–1345 (2008).
- García-Escudero L. A., Gordaliza A., Matrán C., Mayo-Iscar A., Exploring the Number of Groups in Robust Model-Based Clustering, *Statistics and Computing*, 21 (4), 585–599, (2011).
- García-Escudero L., Gordaliza A., Mayo-Iscar A., A constrained robust proposal for mixture modeling avoiding spurious solutions, *Advances in Data Analysis and Classification*, 8 (1), 27–43, (2014).
- Ghahramani Z., Hilton G., The EM algorithm for mixture of factor analyzers, *Technical Report CRG-TR-96-1*, (1997).
- Greselin F., Ingrassia S., Maximum likelihood estimation in constrained parameter spaces for mixtures of factor analyzers, *Statistics and Computing*, 25, 215–226, (2015).
- Hathaway R., A constrained formulation of maximum-likelihood estimation for normal mixture distributions, *The Annals of Statistics*, 13 (2), 795–800, (1985).
- Hennig C., Breakdown points for maximum likelihood-estimators of location-scale mixtures, *Annals of Statistics*, 32, 1313–1340, (2004).
- Ingrassia S., A likelihood-based constrained algorithm for multivariate normal mixture models, *Statistical Methods & Applications*, 13, 151–166, (2004).
- Ingrassia S., Rocci R., Constrained monotone EM algorithms for finite mixture of multivariate Gaussians, *Computational Statistics & Data Analysis*, 51, 5339–5351, (2007).
- Lin T.-I., McNicholas P. D., Ho H. J., Capturing patterns via parsimonious t mixture models, *Statistics & Probability Letters*, 88, 80–87, (2014).
- Maitra R., Clustering Massive Datasets With Application in Software Metrics and Tomography, *Technometrics*, 43 (3), 336–346, (2001).
- Maitra R., Initializing partition-optimization algorithms, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 6 (1) 144–157, (2009).
- McLachlan G. J., Bean R. W., Maximum likelihood estimation of mixtures of t factor analyzers, *Technical Report, University of Queensland*, (2005).

- McLachlan G., Peel D., Robust cluster analysis via mixtures of multivariate t -distributions, in: A. Amin, D. Dori, P. Pudil, H. Freeman (Eds.), *Advances in Pattern Recognition*, vol. 1451 of *Lecture Notes in Computer Science*, Springer Berlin - Heidelberg, 658–666, (1998).
- McLachlan G., Peel D., Mixtures of factor analyzers, in: *Proceedings of the Seventeenth International Conference on Machine Learning*, P. Langley (Ed.), San Francisco: Morgan Kaufmann, 599–606, (2000a).
- McLachlan G. J., Peel D., *Finite Mixture Models*, John Wiley & Sons, New York, (2000b).
- McNicholas P., Murphy T., Parsimonious Gaussian mixture models, *Statistics and Computing*, 18 (3), 285–296, (2008).
- Neykov N., Filzmoser P., Dimova R., Neytchev P., Robust Fitting of Mixtures Using the Trimmed Likelihood Estimator, *Computational Statistics & Data Analysis*, 52 (1), 299–308, (2007).
- R. D. C. Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, URL <http://www.R-project.org>, (2013).
- Rousseeuw P. J., Van Driessen K., A Fast Algorithm for the Minimum Covariance Determinant Estimator, *Technometrics*, 41, 212–223, (1999).
- Steane M. A., McNicholas P. D., R. Y. Yada, Model-based classification via mixtures of multivariate t -factor analyzers, *Communications in Statistics-Simulation and Computation*, 41 (4), 510–523, (2012).
- Stewart C. V., Robust parameter estimation in computer vision, *SIAM review*, 41 (3), 513–537, (1999).
- Tipping M. E., Bishop C. M., Mixtures of probabilistic principal component analyzers, *Neural computation*, 11 (2), 443–482, (1999).